



# INSTITUTO POLITÉCNICO NACIONAL

## ESCUELA SUPERIOR DE CÓMPUTO



### *Tarea 4. Chat Multicast*

#### Alumnos:

- Contreras Barrita José Roberto
- Contreras Mendez Brandon
- Covarrubias Sánchez Daniel

Grupo: 4CV11

Asignatura: Desarrollo De Sistemas Distribuidos

Profesor: Pineda Guerrero Carlos

Fecha: 18-03-22

# Introducción

## Canales Multicast

Los canales multicast son empleados para enviar información a una o más máquinas dentro de un mismo grupo de manera simultánea. Estos canales son bastante empleados en aplicaciones como chats grupales o en servicios de streaming.

Para este tipo de comunicaciones se emplean sockets de datagramas los cuales usan el protocolo UDP (User Datagram Protocol) para el envío de bytes sin necesidad de confirmación de recibido, pero que facilita la transmisión multicast.

## Desarrollo

Para la realización de esta práctica se llevó a cabo la creación de una máquina virtual montada sobre la nube de Azure, con sistema operativo Windows Server 2012. A continuación, mostraremos el proceso para generar la máquina virtual.

Dentro del portal de Azure, seleccionamos el apartado de Máquinas Virtuales dando clic en la sección de Servicios de Azure.

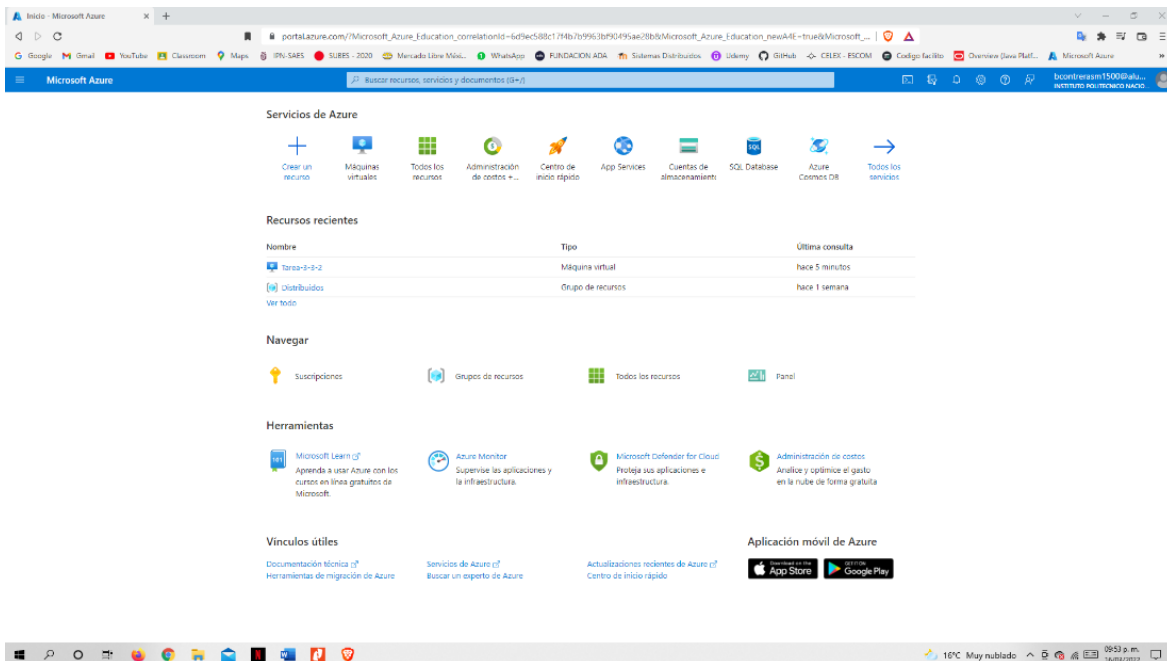
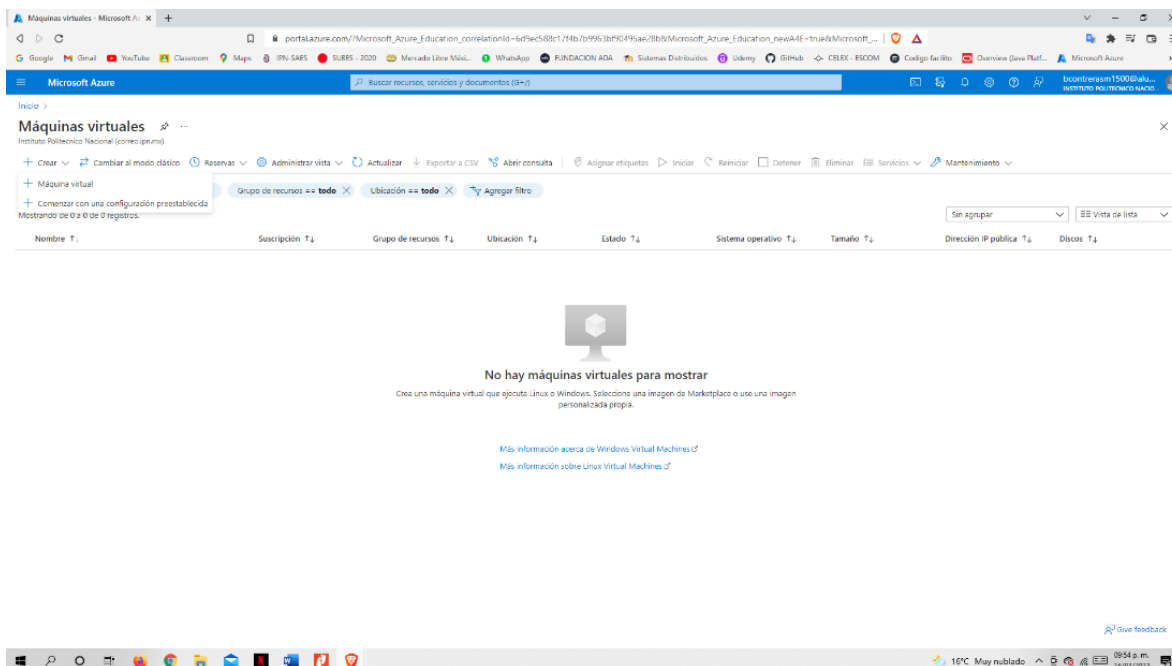


Ilustración 1. Portal de Azure

Dentro de **Máquinas virtuales**, nos dirigimos a la lista desplegable **crear** donde escogeremos la opción de **Máquina virtual**.



*Ilustración 2. Opción para crear máquina virtual*

Comenzaremos a completar el formulario siguiendo las indicaciones publicadas en el Moodle de la materia, seleccionando el Sistema Operativo, la cantidad y tipo de recursos que tendrá la máquina y las reglas de seguridad y acceso, justo como se muestra en las ilustraciones siguientes.

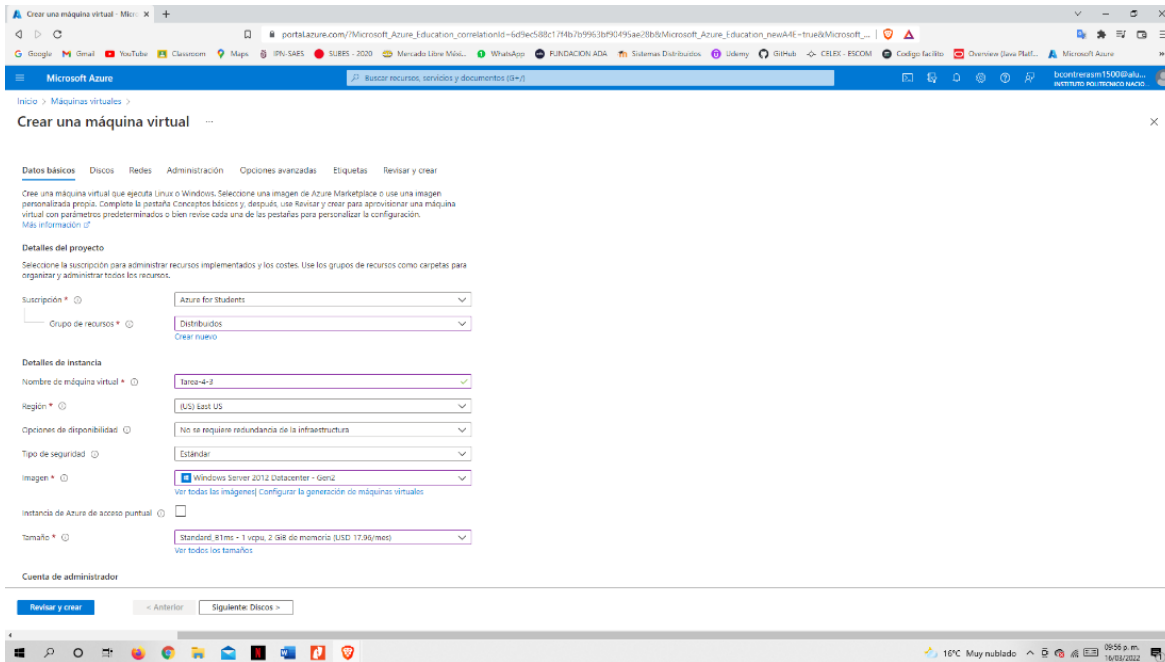


Ilustración 3. Información básica de la VM

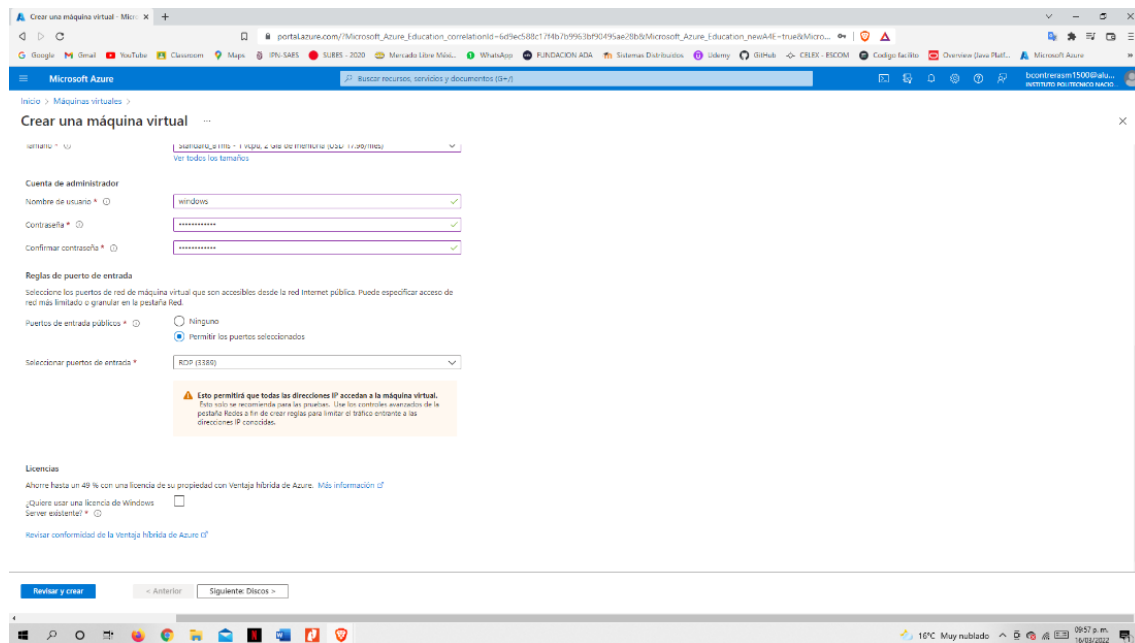


Ilustración 4. Continuación de la información básica

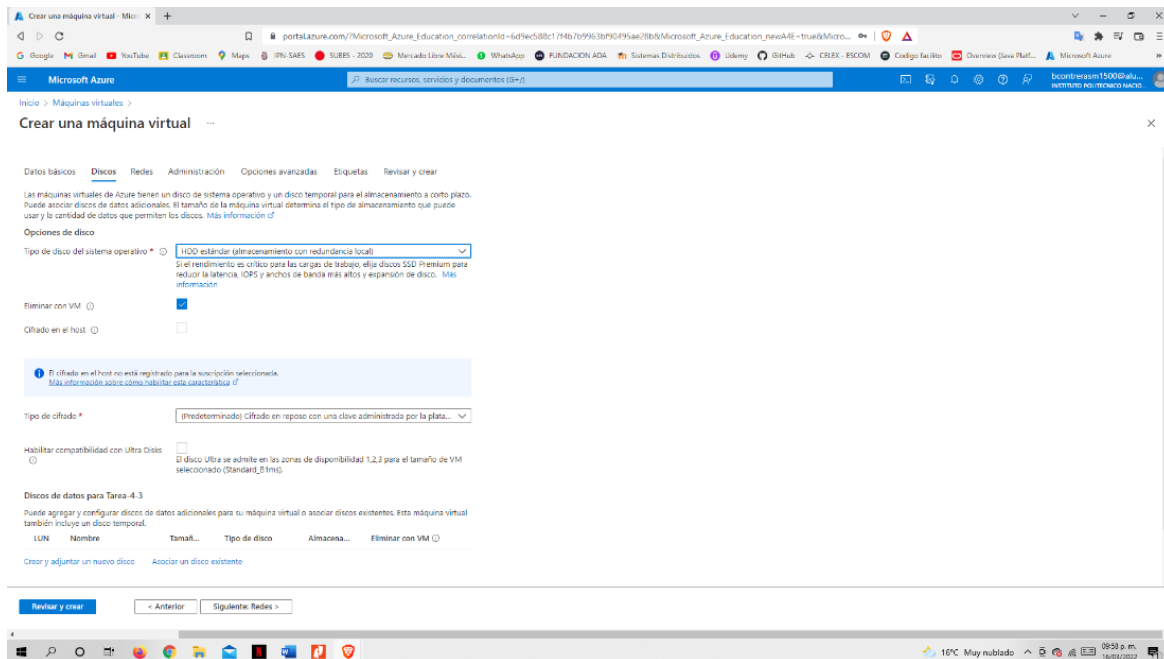


Ilustración 5. Discos de la VM

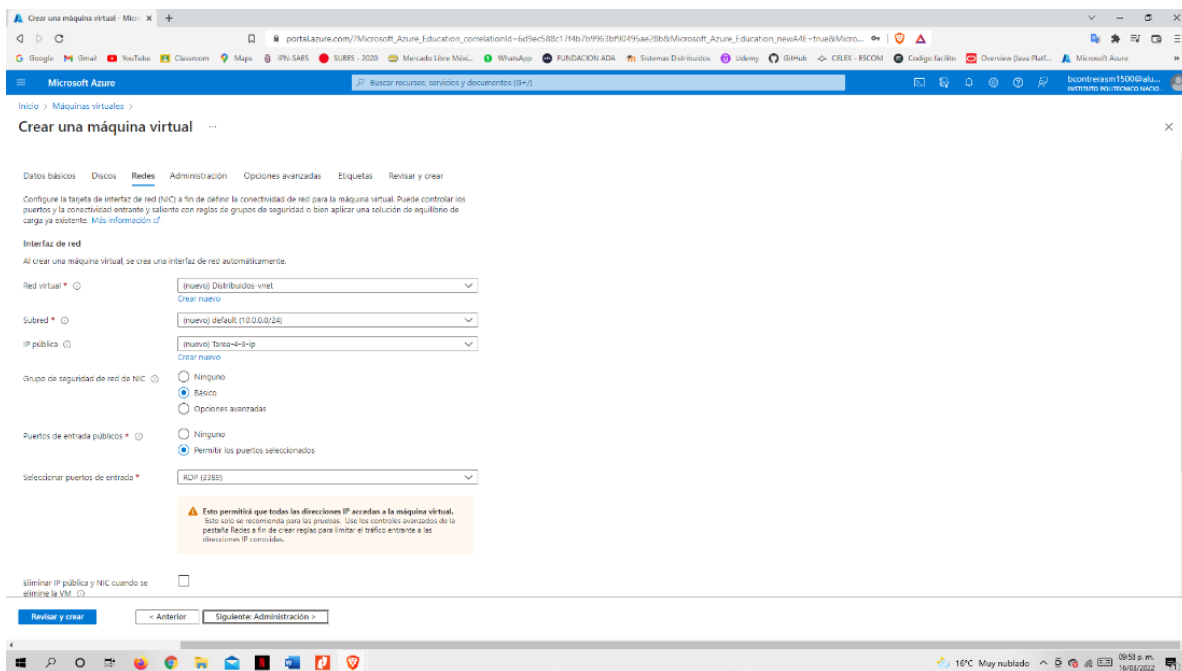


Ilustración 6. Redes de la VM

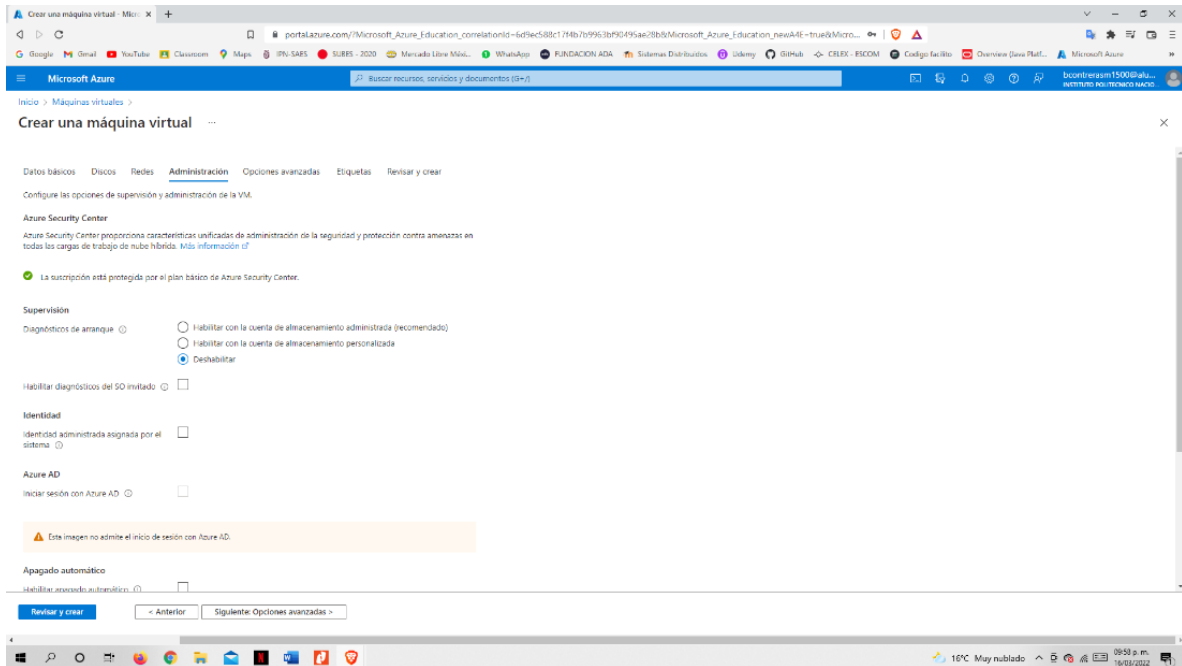


Ilustración 7. Administración de la VM

Finalmente, iremos al apartado **Revisar y Crear**, donde veremos el resumen de lo que será la creación de la máquina virtual y confirmaremos la creación de esta.

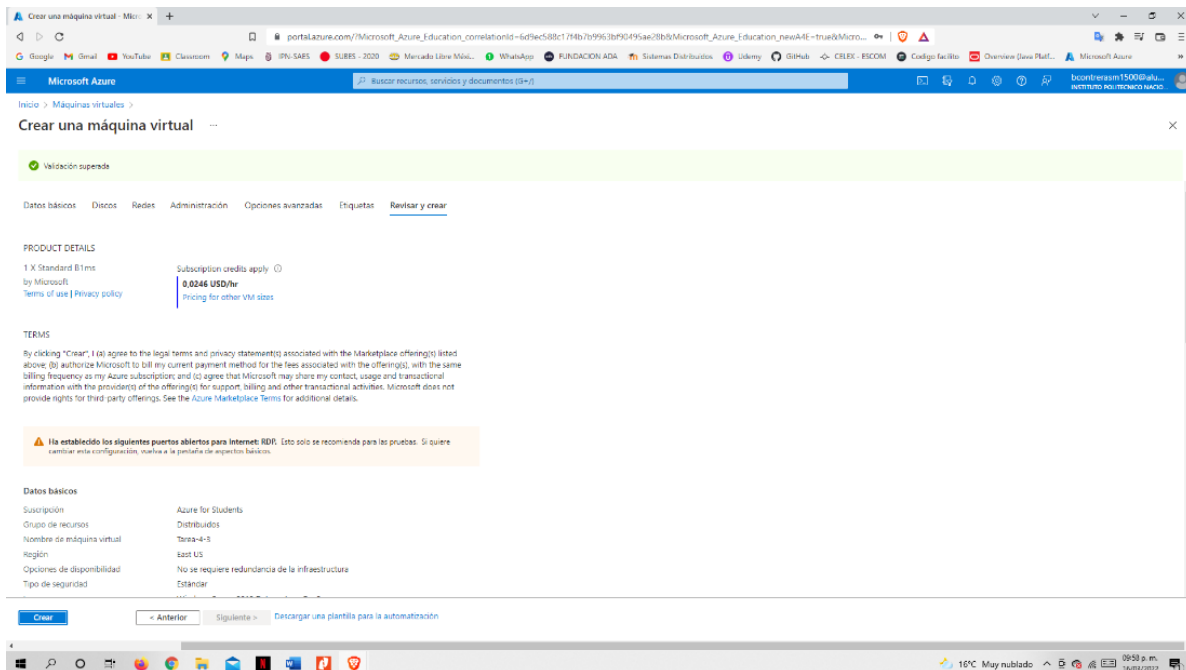


Ilustración 8. Revisar máquina virtual a crear

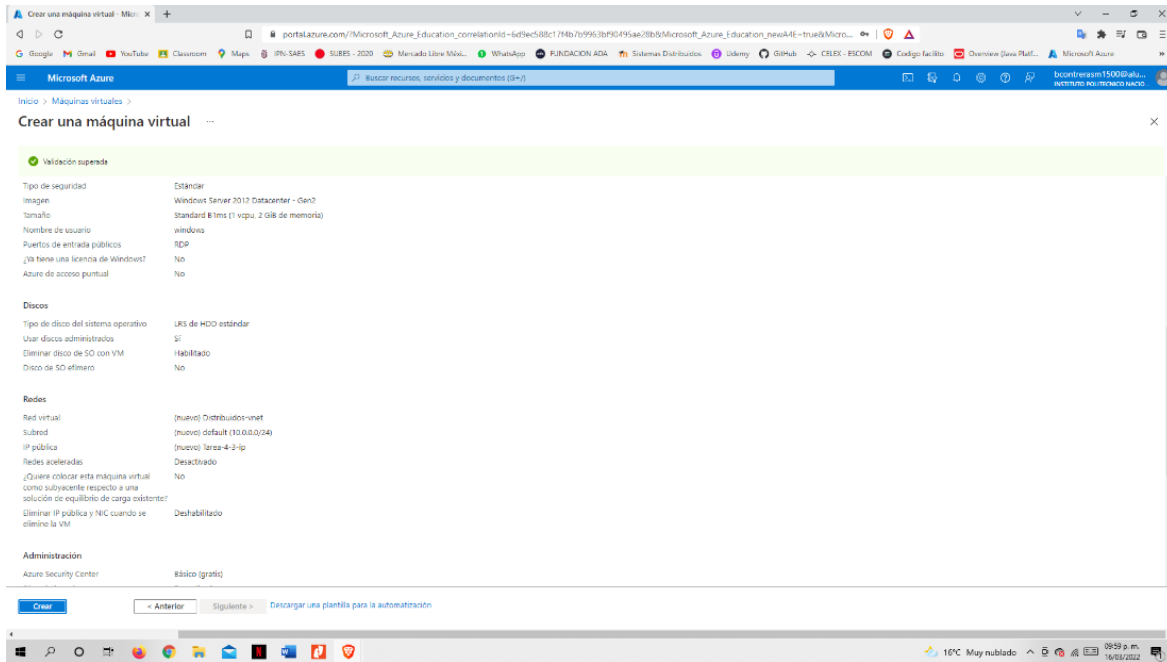


Ilustración 9. Continuación de la revisión

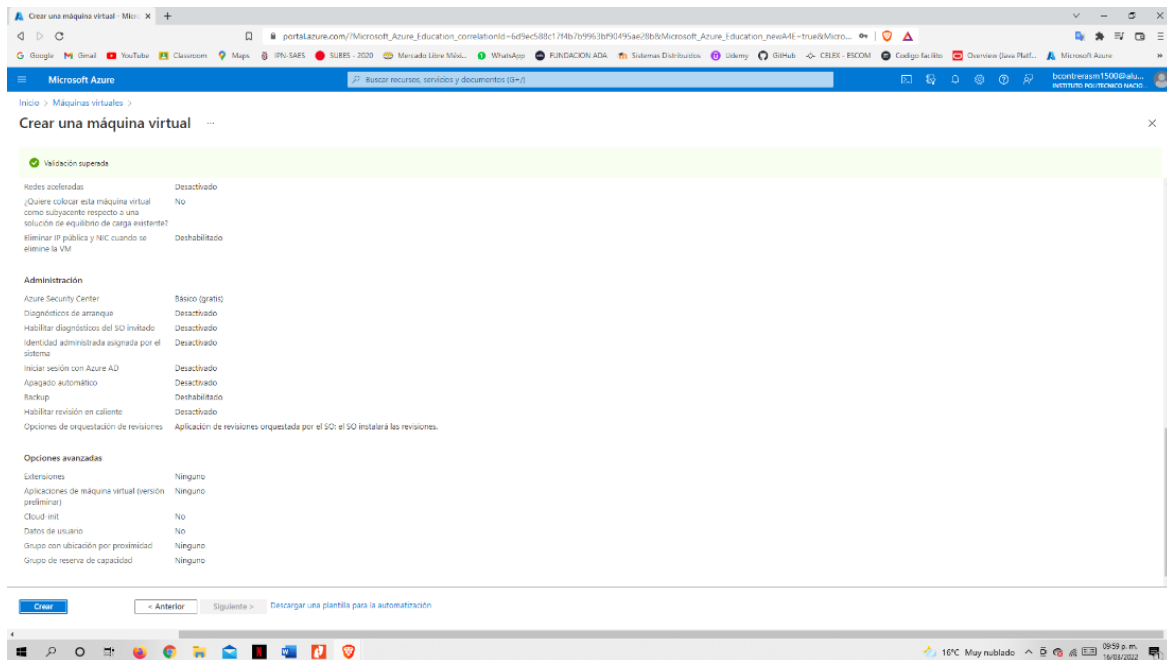


Ilustración 10. Última parte de la revisión

Al confirmar, Azure creará el recurso y nos informará una vez que esté listo para usar.

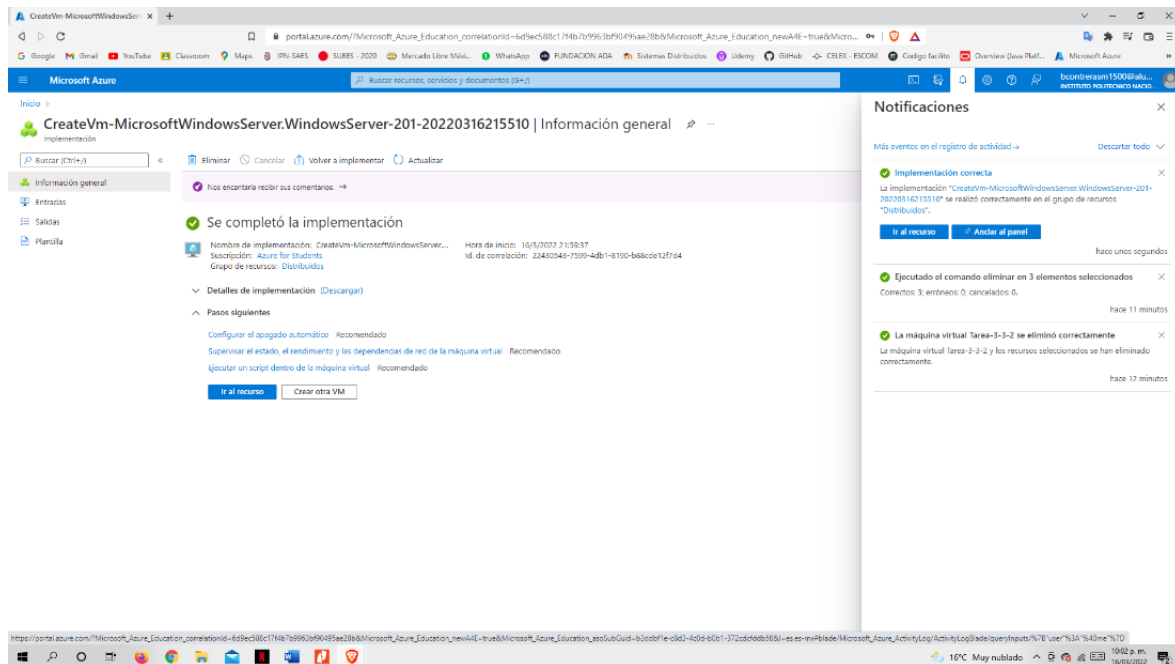


Ilustración 11. Máquina virtual creada

Al tener un sistema operativo de Windows, tendremos que descargar un archivo para la conexión del escritorio remoto y así poder gestionar la máquina y pasar archivos.

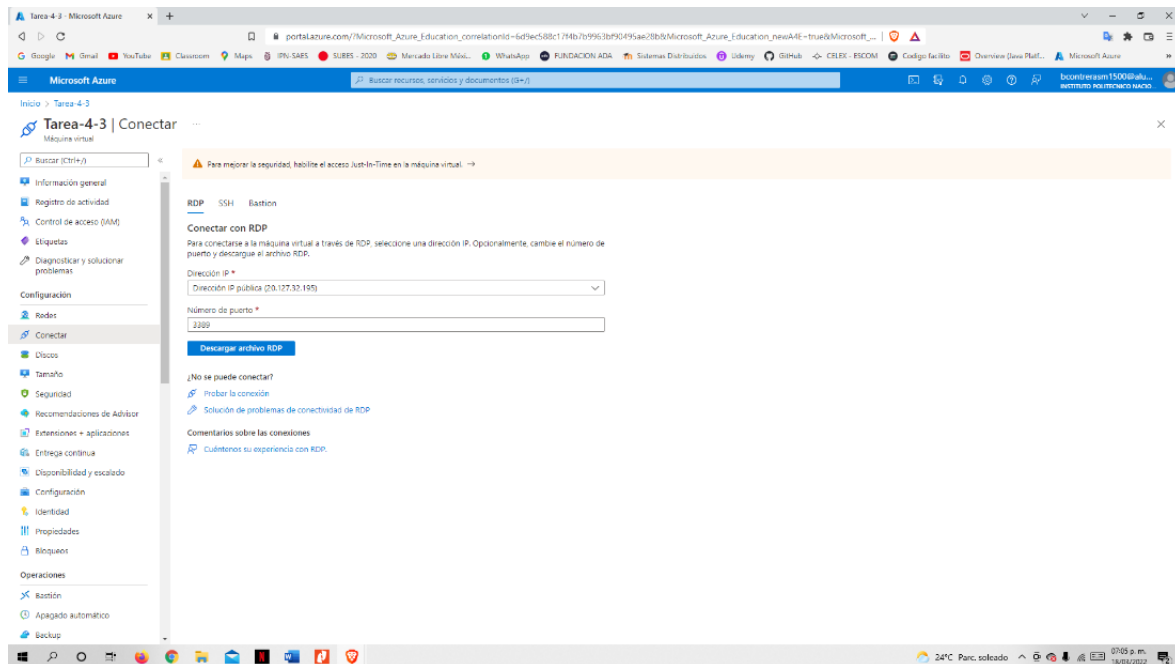


Ilustración 12. Abrir puerto RPD



## Compilación y ejecución

A continuación, se muestra la compilación y ejecución del programa, usando el JDK en la versión 8 de Java, donde se observa la conversación propuesta por el profesor entre los usuarios Pedro, Juan y Ana, con la respectiva codificación de texto “Windows-1252” para que se puedan ver correctamente los signos de interrogación y acentos.

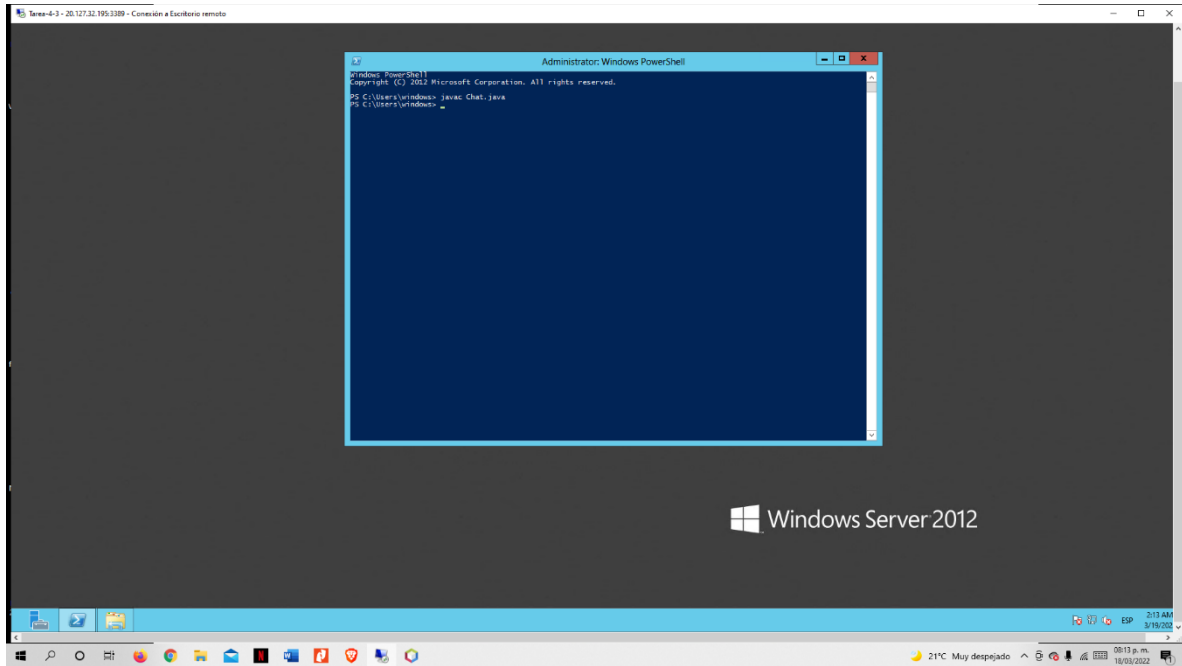


Ilustración 13. Compilación del programa

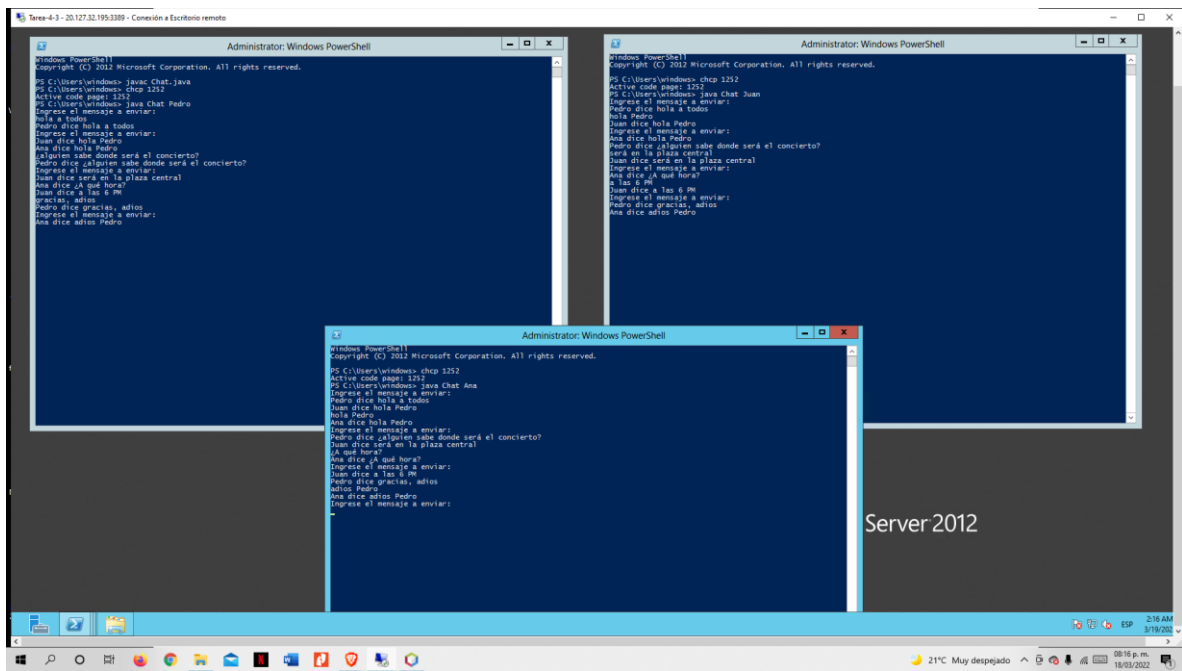


Ilustración 14. Ejecución del programa

## **Conclusiones**

### ***Contreras Barrita José Roberto:***

En esta práctica se implementaron sockets de datagramas los cuales son usados para el envío de audio y video en tiempo real. A pesar de que compañías como Netflix o Twitch usan tecnologías diferentes, la implementación nos permite familiarizarnos con la creación de este tipo de software. En lo personal, el uso de un servidor como lo es Windows Server 2012 fue más complicado que con los servidores basados en Linux.

### ***Contreras Mendez Brandon:***

La realización de esta tarea me permitió conocer cómo crear una máquina virtual de Windows, en este caso, Windows Server 2012, la cual se me hizo un poco más laboriosa de crear y de configurar para poder ejecutar el programa en Java, ya que se tiene que instalar Java, el JDK y poner el debido Path para que se pueda compilar y correr el programa. En cuanto al código no hubo mayor dificultad debido a que ya teníamos los métodos que envían y reciben los mensajes multicast, los cuales se implementan con sockets sin conexión, a través de datagramas, utilizando el protocolo UDP. Además, tanto en el código como en la PowerShell se tuvo que cambiar la codificación de texto, se usó "Windows-1252".

### ***Covarrubias Sánchez Daniel:***

Para esta tarea se vio el envío de mensajes simultáneos a diferentes sistemas por medio de una red multicast, esto con el fin de que todos los usuarios de la red reciban el mismo mensaje y se comuniquen todos al mismo tiempo, cabe mencionar que todos los dispositivos tienen las mismas acciones dentro del sistema. Lo único malo que le veo a esto es que si el usuario no está conectado a la red y se envía un mensaje nunca lo va a recibir.